Document History

Date	State	Stage
25.05.2008	Literature Review	Over all data quality, data profiling and metadata
2.06.2008	Literature Review	Metadata in SQL Server
8.06.2008	Standard Review	Action Plan
12.06.2008	Standard Review	CIHI, New Zealand, Ontario and Statistics Canada Data Quality framework

Table of Contents

TABLE OF CONTENTS....1

ABBREVIATIONS....3

OVERVIEW....3

INTRODUCTION:....4

- Accuracy....5
- Timeliness....6
- Comparability....6
- Usability....7
- Relevance....8
- Ratings....8

DATA PROFILING....9

METADATA....10

- The Metadata Repository:....11
- SQL Server 2005 metadata....12
- Metadata for individual parts of the SQL Server 2005....13
- External Metadata Sources....14

ROADMAP TO METADATA ORIENTED SYSTEM....14

- Metadata Reporting....14

ACTION PLAN....15

DATA QUALITY ARCHITECTURE....16

- Data Quality Profiling steps....17
- Batch Management:....17
  - *Batch Process Metadata:....18*
  - *Extract Repository:....18*
  - *User Feedback Engine:....18*

Batch Manager: *18*
TRANSFORMATION MANAGEMENT: 19
Source absorption: *19*
Data Preparation: *19*
Data Propagation *19*
AUDIT MANAGEMENT 19
Enterprise integrator *20*
Backtrack Manager *20*
ETL Monitor *20*
Predictive Data Manager *20*
BUSINESS RULES ENGINE 21
Error Event Metadata *21*
Column Screening metadata *21*
Structural Screening Metadata *21*
Business Rules Screening Metadata *21*
Error Event Manager *22*
DQ SERVICES 22
DQ Monitoring *22*
DQ Profiling *22*
DQ Reporting *22*
REFERENCES: **23**

Abbreviations

AMO	Analysis Management Objects
BI	Business Intelligence
CIHI	Canadian Institute for Health Informatics
CWM	Common Warehouse Metamodel
DocCenter	SQL Server Business intelligence Documentation Center
DW/BI	Data Warehouse/Business Intelligence
ETL	Extract, Transform and Load
IMS	Information Management System – Database (IBM Technology)
ODS	Operational Data Systems
OMG	Object Management group
PSB	Program Specification Block
SMO	SQL Management Objects
SQL-DMO	SQL Distribution Management Objects
SSIS	SQL Server Integration Services
VSAM	Virtual Storage Access Method (IBM technology)

Overview

This document explains the role and place of metadata in data quality assurance, and describes a methodology and strategy for creating a metadata repository in Microsoft SQL Server 2005.

The document starts with an introduction of data quality, its importance, definition and dimensions. Data quality consists of five different activities, the first of which, data profiling, is examined in further detail. The definition, goals and purposes, along with the three categories of functionality attached to data profiling are then outlined. These include Column, Dependency, and Redundancy profiling. Column profiling provides metadata, which is the main requirement for the successive functionalities.

Metadata is then defined, enumerating most frequently sought types of it, and the methodology and standards for creating an integrated metadata repository are explained.

The next part of the document examines the tools and components of Microsoft SQL Server 2005 to see how each of them handles metadata concepts. As a result, an eight-point strategic plan is adopted from Mundy, Thornthwaite, and Kimball to create a practical system to capture and manage metadata on SQL Server 2005 platform.

A brief description on metadata reporting that explains the tools that would be used to report business and process metadata in SQL Server 2005 follows the strategic plan.

Introduction:

In the present era, data is one of the most important factors for business survival of any industry, and healthcare is no exception. In fact, the drive to reduce cost and the desire to improve patient care makes the healthcare industry one of the most eager customers for clean and integrated organizational data.

But this data needs to posses a number of characteristics if it is to be used for decision-making and strategic operation. The government of British Columbia defines the state of completeness, validity, consistency, timeliness and accuracy that makes data appropriate for a specific use as data quality ()

To build a practical framework for achieving desirable data quality, Strong et al. () suggest taking a customer focused view by treating data processing as a data manufacturing system and to produce data that is "fit to use by data customer". They recognize three fundamental roles: data producers, data custodians and data consumers. Fitness for use would mean that the concept of data quality is relative; therefore dimensions need to be defined to ensure fitness in an integrated health care environment.

Data Producers: Provide Data from Inside or Ouside of the organization

Data Custodians: Code, abstract, verify, validate, aggregate and maintain the data; circulate reports to users

Data Consumers: Use the data for planning, outcome assessments and decisions

Wang and Strong define perceived data quality dimensions in four categories (), as shown in the following table.

Table 1: Perceived Data Quality Dimensions	
Information Quality Category	Information Quality Dimensions
Intrinsic	Accuracy, Objectivity, Believability, Reputation
Accessibility	Access, Security
Contextual	Value-added, Relevancy, Timeliness, Completeness, Amount of Data
Representational	Interpretability, Ease of Understanding, Concise Representation, Consistent Representation
From Wand and Strong ()	

Organizations select from these dimensions based on their own needs and special circumstances. Examples include New Zealand Healthcare data quality framework (6 dimensions), Statistics Canada's Quality Assurance Framework (6 dimensions), Ontario Data Quality Management Framework (4 dimensions) , and CIHI Data Quality Framework, which will be presented here in further detail.

CIHI has been publishing a framework for data quality since 2000. the latest revision, 2006, has selected five of these dimensions to implement a data quality assessment tools ().
CIHI selects Accuracy, Timeliness, Comparability, Usability, and Relevance. They then divide these dimensions into characteristics, and define a set of criteria for each characteristic. Using a ranking system of met, unmet, and unkown or not applicable, the strengths and limitations of data is assessed, and areas of intervention identified. To help prioritize those intervention activities, some of the criteria follow another ranking system of minimal, moderate and significant.

CIHI defines the roles and responsibilities of data quality framework as follows:

- Senior Management
    - Provide Resource, Support DQ in all new initiative
- Product Areas
    - Evaluate data quality and address issues
    - Document quality
    - Conduct studies and identify ways of improvement
- Data Quality Section
    - Provide guidance, review and report on compliance
    - Assist in studies
    - Conduct R&D on data quality
    - Train and update framework annually

Accuracy

The accuracy dimension refers to how well information in or derived from the database or registry reflects the reality it was designed to measure.
Accuracy is what most people think of when they think of data quality. Accuracy refers to how well information in or derived from the database or registry reflects the reality it was designed to measure. When people ask if an estimate is good, most of the time they are asking if the estimate is accurate.
The accuracy of a database depends on many factors, some of which are difficult to measure. When considering accuracy, it is important to keep the following in mind:

- Are all the appropriate data present?
    - Coverage: do you know who should be submitting data?
    - Unit non-response: have all the records been submitted?
    - Item non-response: are the submitted records complete?
- How good are the data?
    - Capture and collection: what measures exist to minimize error?
    - Measurement error: how well were the data reported to CIHI?
- What is done with the data?
    - Edit and imputation: are the checks and modifications to the data logical and consistent?

- Estimation and processing: are the processes used to generate values documented and tested?

Timeliness

Timeliness refers primarily to how current or up to date the data are at the time of release, by measuring the gap between the end of the reference period to which the data pertain and the date on which the data become available to users.

Timeliness refers primarily to how up to date the data are at the time of release. How current the data are is measured in terms of the gap between the end of the reference period to which the data pertain and the date on which the data become available to users. Timeliness is therefore closely associated with relevance, in that if this delay is too great, the data may no longer be relevant for the needs of users. Though data must be produced in time to assure relevance, acceptable timelines may vary across CIHI data holdings. More complicated databases or registries (that is, databases capturing longitudinal data), or those that are dependent on other databases for data, cannot be held to the same timelines.

If too much emphasis is placed on considerations of timeliness, accuracy may be compromised. For example, without sufficient time for clinicians to complete hospital charts, hospital discharge data might be timely but incomplete. It might be argued that this sacrifice in completeness is not worth the gain in timeliness. Sufficient time must also be set aside after the database or registry year-end close and prior to release in order to check the data and to document the limitations for users. As there is always more quality control and documentation that can be done, a balance must be struck between timeliness and accuracy. At a minimum, the recommended data quality documentation must be available in time for release.

The purpose of the timeliness dimension is to examine how current the data are and whether the recommended data quality documentation was made available in time for release. The criteria in this dimension also assess whether major database or registry reports are released on schedule. The dimension is comprised of the following characteristics:

- Data currency at the time of release: are the data made available in a reasonable amount of time?
- Documentation currency: are key documents released on time?

Comparability

The comparability dimension refers to the extent to which databases are consistent over time and use standard conventions (such as data elements or reporting periods), making them similar to other databases.

Within an organization like CIHI, with many different databases, comparability facilitates the understanding, interpretation and maintenance of the data. It is also directly related to the portion of CIHI's mandate that applies to the development and maintenance of a comprehensive and integrated health information system.

Databases that are comparable will use the same data definitions, collect similar types of data and have the potential for record linkage with other similar databases. This in turn makes it possible to combine data from disparate sources in order to address important research questions that cannot otherwise be examined. Research on continuity of care is a prime example, given the range of clinical databases required for analysis (everything from emergency care to chronic care).

An additional advantage of comparability is that it can be used to assess other aspects of data quality, such as accuracy. Comparison of similar data systems can be an effective way of examining issues of coverage, coding errors and non-response.

The comparability dimension tells us how well databases meet a common standard. It is comprised of the following characteristics:

- Data Dictionary standards: does the database use CIHI standards for data definitions?
- Standardization: can common groupings be derived from the data?
- Linkage: can databases be joined by a common data element?
- Equivalency: are data values being converted correctly?
- Historical comparability: are data comparable over time?

Usability

Usability reflects the ease with which a database or registry's data may be understood and accessed.

If data or other information products are difficult to use, they can be rendered worthless no matter how accurate, timely, comparable, or relevant they may be.

Several factors contribute to the usability of a database's or registry's data. In general, the greater the number of limitations or exceptions associated with the data, the more difficult the data will be to interpret. Efforts made to reduce or prevent data quality limitations or to improve the standardization of data improve not only the ease with which the data can be used, but also the accuracy of the data.

Inconsistent database or registry methods may also complicate interpretation. The benefits derived from the introduction of new methods (for example, data element name or definition changes) should therefore be weighed against any loss in interpretability. Simply put, the fewer the limitations and changes, the easier the data will be to interpret.

To aid in the interpretation of the data, key users should be informed of any known major limitations at year-end and on an ongoing basis after release.

Once major limitations are known, they should be documented for users. Database or registry methods and changes to the methods should also be documented for users. Also, the data have to be in a readily accessible user-friendly form. Finally, no matter how well documented or accessible, if users are not aware of a database's or registry's existence, the data will not be used.

The purpose of the usability dimension is to identify problematic aspects of a database or registry that are related to the interpretability of its data, as well as to identify how well documented and accessible the data are. It is comprised of the following characteristics:

- Accessibility: how readily accessible are the data?
- Documentation: how well documented are the data?
- Interpretability: how easy is it to understand the data?

Relevance

Relevance reflects the degree to which a database or registry meets the current and potential future needs of users.
Maintaining relevance requires keeping in touch with key users and stakeholders.
Relevance is concerned with whether the available data inform the issues most important to users. In addition to ensuring that its data are accurate, timely, comparable and usable, to fulfill its mandate CIHI must also make certain that its data holdings continuously reflect Canada's most important health care information needs.
The challenge is to balance the differing needs of current and potential users to produce a program that goes as far as possible in satisfying key needs.
The purpose of the relevance dimension is to assess how well a database or registry can adapt to change and whether the database or registry is perceived to be valuable. It is comprised of the following characteristics:

- Adaptability: can user needs be anticipated and planned for?
- Value: how valuable are the data?

Ratings

Ratings are used in the assessment tool as a guide to highlight strengths and identify weaknesses and limitations of a database or registry.
Ratings are only one component for product areas to consider in determining fitness for use. The subjective nature of data quality and the differing nature of databases and registries mean that no rating system will be able to identify all problems with data quality.
Scores which do not identify data quality issues do not necessarily mean a database or registry's data are problem free. Similarly, scores that indicate some data quality issues do not necessarily mean that a database or registry's data should not be used. It is the responsibility of the product area to identify in the presence or absence of data quality issues the strengths and weaknesses of the database or registry.
In most cases, each criterion is given a rating of either *met*, *not met*, *unknown* or *not applicable*. In select cases, criteria are rated according to other predetermined categories: *minimal or none*, *moderate*, *significant* or *unknown*. Each criterion has a statement or data table that can be used to determine what rating should be assigned.
These scores are for internal purposes only and should appear only in the evaluation report. They should not be included in any external documentation.
All assignments are made together with staff responsible for each database or registry.
Determination of the level of quality must be done after confidentiality, privacy and security considerations have been addressed (if necessary).

Some of the activities in data quality include ():

- Data profiling - initially assessing the data to understand its quality challenges
- Data standardization - a business rules engine that ensures that data conforms to quality rules

- Matching or Linking - a way to compare data so that similar, but slightly different records can be aligned. Matching may use "fuzzy logic" to find duplicates in the data. It often recognizes that 'Bob' and 'Robert' may be the same individual. It might be able to manage 'householding', or finding links between husband and wife at the same address, for example. Finally, it often can build a 'best of breed' record, taking the best components from multiple data sources and building a single super-record.
- Monitoring - keeping track of data quality over time and reporting variations in the quality of data. Software can also auto-correct the variations based on pre-defined business rules.
- Batch and Real time - Once the data is initially cleansed (batch), companies often want to build the processes into enterprise applications to keep it clean.

There are many technologies that apply to creating and maintaining an effective data quality assurance program, the most important of which include data profiling, data cleansing, metadata repositories, data filtering and data monitoring.

Data Profiling

The first step of data quality, data profiling, is considered the most important technology available to the data quality assurance team (). It is defined as the application of data analysis techniques to existing data sources for the purpose of determining the actual content, structure and quality of data. That is in contrast to the data analysis techniques deriving business information from data ().

The purpose of these data analysis techniques include ():

- Find out whether existing data can easily be used for other purposes
- Give metrics on data quality including whether the data conforms to company standards
- Assess the risk involved in integrating data for new applications, including the challenges of joins
- Track data quality
- Assess whether metadata (what correct data is) accurately describes the actual values in the source database
- Understanding data challenges early in any data intensive project, so that late project surprises are avoided. Finding data problems late in the project can incur time delays and project cost overruns.
- Have an enterprise view of all data, for uses such as Master Data Management where key data is needed, or Data governance for improving data quality

The main goal of data profiling is to generate accurate metadata by reverse-engineering the current data. However, it cannot find all inaccurate data, but just rule violations where data and metadata do not agree. This includes invalid values or violation of data rules or data structure, but may still miss wrong data in some instances ().

Inputs to data profiling consist of existing metadata, which can be accurate or not, and existing data, which again can be inaccurate or accurate. Outputs include accurate metadata and facts about inaccurate data. Wherever data and metadata do not agree, a

decision is required as to which should be corrected. There might be cases, however, that they are both wrong and agree with each other. Various analysis techniques are defined to capture such instances, but still there might be cases that skip detection.

The Data profiling functionality is often divided into three categories. The names for these categories often differ depending on the vendor, but the overall process is in three steps, which must be executed in order (23):

- Column Profiling (Including the statistics), which provides critical metadata, required for the next steps.
- Dependency Profiling, which identifies intra-table dependencies. Dependency profiling is related to the normalization of a data source, and addresses whether or not there are non-key attributes that determine or are dependent on other non-key attributes. The existence of transitive dependencies here may be evidence of second-normal form.
- Redundancy Profiling, which identifies overlapping values between tables. This is typically used to identify candidate foreign keys within tables, to validate attributes that should be foreign keys (but that may not have constraints to enforce integrity), and to identify other areas of data redundancy. Example: redundancy analysis could provide the analyst with the fact that the ZIP field in table A contained the same values as the ZIP_CODE field in table B, 80% of the time.

Metadata

Metadata is defined as data about data. It describes the rules for correctness of the data, and therefore is essential for determining inaccuracies ().

Typical types of metadata sought are (23):

- Domain: whether the data in the column conforms to the defined values or range of values it is expected to take
    - For example: ages of patients in pediatric ward are expected to be between 1 and 15. An age of 17 would be considered out of domain
    - A code for flammable materials is expected to be A, B or C. A code of 3 would be considered out of domain.
- Type: Alphabetic or numeric
- Pattern: a North American phone number should be (999) 999-9999
- Frequency counts: most of our customers should be in British Columbia; so the largest number of occurrences of state code should be BC
    - Statistics:
        - Minimum, maximum, mean (average), median, mode values and standard deviation
- Interdependency:
    - Within a table: the zip code field always depends on the country code.
    - Between tables: the patient number on an order should always appear in the patients table

The process and methodology of making a metadata repository is controversial and relatively uncharted (23). Metadata is divided to “front-room” and “back-room”, to distinguish between the aims of ‘running queries and reports smoothly’ and ‘guiding extraction, cleaning and loading processes’ respectively ().

More recently, Kimball has divided metadata into three major categories: business, technical, and process. He then proceeds to describe the Common Warehouse Metamodel, the industry standard for storing and exchanging metadata (). In this sense, metadata describes the contents of warehouse, the structure that holds them, and the process that brought them into being. Nevertheless, there are many overlaps between these types of metadata, sometimes serving as causes for confusion. Security-related metadata, for example, is both technical (since the system needs to identify the users, their groups, and access rights), and business (since the users need to know what data is available to access). Most elements, however, can be placed in one category only.

The purpose of metadata is either defining a process, object, or behavior, or describing an object or a process. The first concept has been around for a long time in the form of configuration files that change the properties and behavior of an object without need to change in the code. The second concept merely documents the target and leads to clear understanding of it. Changing the configuration file will change the behavior, but to prevent misconception, the documentation should change as well.

Technical Metadata, primarily definitional, defines the objects and processes that makeup the data warehouse itself. Tables, Fields, data types, indices, dimensions, measures, and partitions are described here. In ETL process, technical metadata describes the sources and targets for a particular process, the transformation (business rules and data quality checks), and the tasks that makeup the process and their schedule, and so on.

Business Metadata, primarily descriptive, tells the users what data they have, where it comes from, what it means and what are its relationships to other data in the system. In SQL Server 2005 Analysis Services, Data source views, name and description fields, and AttributeHierarchyDisplayFolder properties serve that purpose.

Process Metadata, again primarily descriptive, logs the resource consumption during system operation. In ETL process, each task logs its execution (start time, end time, disk read, disk write, rows processed, etc) that eases troubleshooting the ETL or query process. As people use the system this data becomes a critical input to the performance monitoring and improvement process. When these activities are the main responsibility of the organization, i.e. the organization is a DW/BI team, then the process metadata becomes their business metadata for that organization.

The Metadata Repository:

Many tools for analysis and reporting use Metadata information. Therefore, there should be a shared place where this data is stored, and a standard of saving the metadata so that these tools can access and use it. There might be need to describe this metadata when it is

saved in propriety XML files in order for other tools to understand and import it, and this is when there would be metadata on metadata and that is what makes the build of metadata repositories a significant effort.

The effort, Nonetheless, is worth because having a standard, shared repository for metadata has many advantages. Tools can exchange and reuse metadata, and new steps can be defined with the least risks. Front-end tools can be replaced with better ones, even from a different vendor. There would exist a single, central, official and active documentation on the data warehouse sporting a common understanding of the contents and structure of the data warehouse. And, last but not least, an easier way to assess changes through impact (what down stream object will be affected by a given change) and linage (how a certain element came into being) analysis.

Since 2001, OMG (object management group) has published Common Warehouse Metamodel (CWM). This is a standard framework for data warehouse oriented metadata. The overall standard is composed of three parts: CWM is the domain model with over a dozen sub-domains (or sub-metamodels) including ones to describe relations, transformations, OLAP, and front end components. The second part of standard, MOF (the Metadata Object Facility), is the language to define a specific instance of CWM. Finally, the XMI (XML Metadata Interchange) is the structure to support the interchange of metadata using XML messages.

Vendors, however, have been slow to adopt CWM, claiming it to be not flexible or comprehensive enough. Metadata managers need to unravel a couple of thousands of pages of documentation that is based on CORBA and UML. CWM compliant products sometimes depend so much on extended, custom structures in the CWM that makes their output unusable by other tools. OMG has been very slow in updating and shows little activity on the CWM standard. The concept, however, remains valuable.

SQL Server 2005 metadata

While SQL Server 2005 does not support CWM, its toolset is mostly metadata driven. Metadata in the relational engine is stored in tables and managed by stored procedures, while Analysis Services and Integration Services employ XML files to store metadata in an object-oriented structure, and use AMO (Analysis Management Objects) or SMO (SQL Management Objects) to access and manage them.

Since these metadata are not integrated across the tools, there needs to be a detailed inventory of metadata available, the ones actually being used, as well as the tools for viewing and/or managing them.

SQL Server 2005 has a set of components that support more than one tool. The first one is the SQL Server Agent, which is the job scheduler and contains information on jobs, steps and schedules. Its metadata can be accessed through stored procedures, system tables, SMO and SQL Server Management Studio.

SQL Server Profiler is the activity-monitoring tool. It places a trace on a specific event and tracks its occurrence, writing the results to a table or file. Its metadata can be accessed using stored procedures or Profiler Tool.

SQL Server BI Documentation Center (DocCenter) is a reporting utility that gathers and explores metadata from three major sources, namely the relational engine, the Analysis Services, and the Integration Services. It creates a set of XML documents with an XSLT reader, which very useful in the survey step of creating a metadata strategy.

Metadata Reporting Pack for SQL Server Integration Services (Metadata Pack) parses the Integration Services to identify the sources, targets, transformations, and dependencies in data flows and stores them in a relational database for impact and linage analysis.

Metadata for individual parts of the SQL Server 2005

The SQL Server Relational Engine has system tables that can be assessed through catalog views, stored procedures, SMO and SQL-DMO (SQL Distributed Management Objects). Descriptive information, such as business metadata is stored along the table and column definitions using the extended properties function. The SQL Server Profiler described above can capture process metadata for relational engine.

Analysis Services stores metadata through its object model. This model can be accessed with the BI development studio and the SQL Server Management studio. Dot Net based applications can use AMO to access this object model, and the process metadata is again captured through the SQL Server Profiler. Analysis Services offers Perspectives, which provide for logical sub-setting of a cube and are used in Report Builder models, as well as its own security metadata to limit access to data subsets.

The Data Source Views (DSVs) are used by Analysis Services, Report Builder and Integration Services and provide an abstraction layer between the source systems and the processes. DSV has a sub-model named a diagram, which simplifies it and supports the idea of role-playing tables. However, DSV is not directly referenced as metadata outside BI Development Studio.

Integration Services impose the challenge of source tracking, since they are essentially a visual programming environment as apposed to a structured database, and naturally have their own object model that is actually a result of naming conventions for packages and tasks. The Process metadata, fortunately, still is captured by the SQL Server Profiler and used to track the performance of SSIS packages. Users can also get a sense of where data came from by examining the Audit dimension and associated audit tables that show Audit system tying the process metadata back to the actual data that was loaded in a given ETL package.

Reporting Services is an example of SQL Services that is entirely metadata driven. The metadata for contents, operation, usage and security are kept in tables in the Report Server database. It also exposes an object model that is accessible through a web service.

The Report Builder Model is created using the BI Development Studio and has its own DSV. Moreover, these models can be built directly from Analysis Services cubes.

External Metadata Sources

Metadata for accessing information from the operating system or computing environment are also useful when implementing business intelligence in an organization. Chief among these are the System Monitor Tool, which defines traces for a vast number of system events, and the Active Directory, which is the network-based user logon management facility. Beyond that, data from other management applications (e.g. source control tools such as Visual Source Safe) are treated as metadata too.

Roadmap to Metadata oriented system

While vendors have been building metadata repositories for decades, there are very few examples of large-scale, robust, successful metadata systems. The major effort is to figure out what metadata to capture, where to capture it, how to integrate it, how to use it and how to keep it synchronized and maintained. This makes it easy to get trapped in the metadata morass. The key is to avoid over-engineering metadata. That starts with a correct, complete, well-maintained and accessible business metadata.

The strategy to create metadata begins by creating an inventory of the existing metadata, showing where they are, where they came from, who owns them, how to view and change them, and where and how to use them. The next step is to identify or define metadata that needs to be captured, and to decide on the definitive location where each element will be stored and edited. Once this is done, it’s time to create systems to capture metadata that does not have a home. There should also be programs or tools to share and synchronize metadata between its master location and any subsystem that needs them.

The strategy should also include training the team and the users about the importance of metadata and the assignment of metadata creation and updating responsibilities. There should be some metadata access tools that users can employ to browse the repository to find out what is available in the BI system. Finally, the metadata should be managed and its usage monitored, keeping it complete and current.

Metadata Reporting

The easiest tool to deliver business metadata is the Analysis Services. The Perspective feature in Analysis Services provides a sub-setting capability, where users can see fact tables and associated schemas for a particular subject area. It can include the description of cubes, dimension, attributes, and even user rights. While Report Services could use this metadata to some extent in Report Builder, it does a poor job of providing the user with the ability to explore the contents of the warehouse.

The Metadata stored in the Analysis Services can be viewed directly with the AMO Browser that ships as a sample program with SQL Server 2005, or by a custom .Net application that uses the AMO component to delve into the metadata repository of the analysis services.

For Metadata outside Analysis Services, one needs to go back to the relational model and provide a set of reports for the users to explore the metadata in the extended properties. It can then be integrated with the Analysis Services metadata by creating a business metadata schema. This schema can now allow for additional metadata fields that may not exist elsewhere.

This schema is essentially a hierarchy starting from a database table, which contains zero to many subject areas, each containing zero to many objects, each of which consisting of zero to many attributes. There Join tables than allow for one subject area to be mapped into several databases, one object to be present in multiple subject areas and one attribute to be part of several objects.
Process Metadata Reporting gives the user a sense of what is really happening in the system and helps the technical team to respond appropriately to the situations that come up during system operation. The SQL Profiler is a good tool in this area and can capture current activity and create a log file of active processes in SQL Server and Analysis Services. The SQL Management studio can also view the current Integration Services activity. Reporting Services has its own execution log schema that can be added to the Performance tool.

All the process metadata reports should publish to the team website. This supports the idea of a single place for business information, even for the DW/BI team. Tools that follow the trends and behavior of these logs over time are critical in warehouse management, performance tuning, long-term capacity planning and educating management.

Action plan

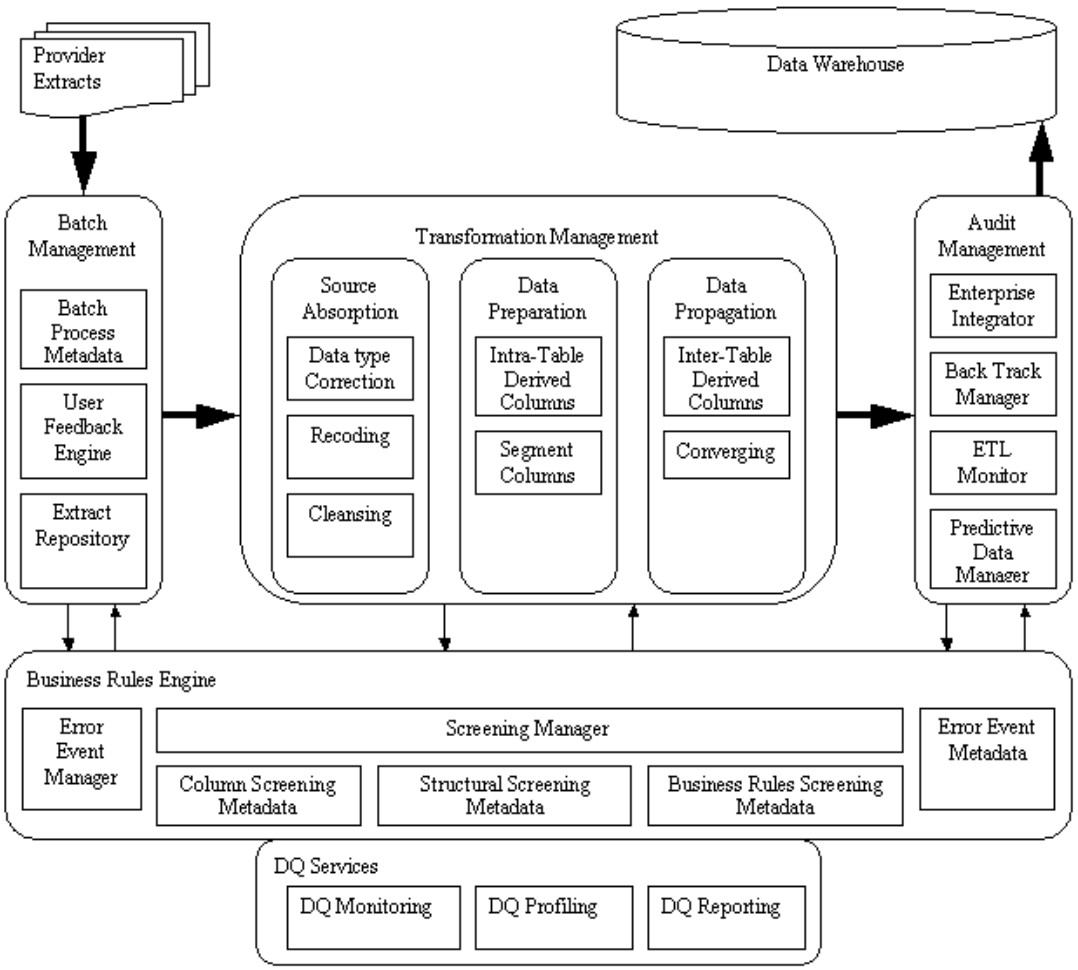
- A common format and in a common place should be selected to record the gathered Information. This will facilitate dealing with multiple data sources. A protocol of verification between technical staff and stakeholders is put in place to determine the completeness of data gathering.
- The source of information starts with the data source, the database or file it is stored in, and how to access the information, which should be as unfettered and unfiltered as possible.
- The next step is to collect metadata on the column-level layout of the data source. In a relational database system, this can be extracted from catalogs. For IMS or VSAM data sources, COBOL copybooks or PL/1 include files contain this information.

- Structural information is the next topic of interest. The referential constraint information, triggers, stored procedures, cull data filtering, and validation rules fall in this category. For IMS, the PSB (program specification block) should be consulted in this regard.
- Application interfaces, existing metadata, descriptive documentation, and source code of programs that insert, delete and update the tables are other sources of insight about the data. It must be noted that sometimes data rules are enforced through human procedures instead of software programs. Personnel manuals used by data entry personnel may contain valuable rules.
- Sample of data is extracted from each source. This serves as the proof of concept showing the data profiling process
- A baseline profile is constructed, showing for each applicable column, the expected values such as mean, sigma, maximum and minimum values, etc. this will be visited periodically to reflect the true nature of data
- An SSIS package is developed to usher the ETL process from source to destination. This SSIS package has two main responsibilities
    - It should record all the errors and exceptions that occur during the ETL process, such as a data element not being of expected type, that results in exclusion of that row of data from the destination cube. There are also some checks that if satisfied, will result in change or nullification of data. These should also be logged. Furthermore, the package should freeze a micro subset of data (criteria 46) and save the original row data in a secure position (criteria 25).
    - It should provide a profile and a "DQSnap" of the extract, identifying the number of input rows, number processed, characteristic (mean, standard deviation, etc) of each column were applicable, and so on. These data qualify the current extract and are used to update future revisions of the baseline
- Through discovery and backtracking operations, the baseline is updated and the tests to check for suspicious data are developed.
- Periodic quality assessment reports reflect the results of the quality assessment for each extract and each source and are produced at the same time the ETL process is complete. They follow a customized format based on CIHI template.
- Data quality documentation for users is maintained and published with annual revisions
- Data quality methods documentation is maintained for each source that documents the data flow and ETL process for that source and includes all methodological information so that a person new to subject could take over the project if there was loss of staff

Data Quality Architecture

Data Quality Profiling steps

According to Olson (), there are 5 major steps in data profiling methodology. These were adapted in Kimball's data quality architecture() to add error and audit dimensions to ETL systems.



The Kimball architecture for data quality aims to design an easily implemented, non-disruptive, scalable, and comprehensive foundation for capturing data quality events as well as measuring and ultimately controlling data quality in the data warehouse (). This architecture includes a number of interacting layers that encapsulate operations and categorize functionality.

Batch Management:

This group implements the interface to the providers of the information. This is a key component to move from an extract-base towards a real-time ETL system that is independent of how the input data is provided. the members of this group include:

Batch Process Metadata:

This is a repository describing who the providers are, what are the inputs, how these inputs are provided, which process step to trigger when they arrive, and the feedback and the reporting information about them.
As an example, consider a data source, which consists of three flat text file extracts that are placed in a specific directory everyday. The batch process metadata shows this information, along with the data about the person to whom the system shall report if it can't find one of these extracts when expected. Also, it tells the system which ETL process step to execute once these extracts arrive, and what information to record about the ETL process that these extracts went through. If, for example, in a certain condition an email is to be send to someone, the Batch process metadata contains the template and the contact info for the recipient.

Extract Repository:

According to CIHI, one of the criteria for the accuracy dimension is that the raw data be stored in a secure location (). The extract repository is in charge of that. It will save the extract and restricts and logs access to them. It also manages the life cycle of input data, deleting the old data when it is no longer of significance.
In addition to storing and managing the life cycle of the extracts, the extract repository also records the attributes for each batch that the ETL system processes. Examples of such attributes include, as stated by Kimball (), The batch date-time stamp, start and end time stamps, total number of records processed, total number of screen tests performed, total number of errors encountered, resource contention, and maximum error severity once the batch was run.
In case of real-time input, where data arrives in individual rows as they are produced in the ODS, the extract repository will group the records in a time frame as a virtual batch to record their attributes and perform life cycle management. The issue of exact implementation detail of such functionality is left open for future versions of architecture, but the functionality itself remains the responsibility of the extract repository.

User Feedback Engine:

As its name implies, this component provides feedback to the suppliers and ETL managers about the quality of the source input data. It is built into the organizations web portal, managing access levels for different users, so that users only have the right to see certain portions of the feedback data that is relevant to them, while managers and administrators have the right to change the metadata on which the batch management operates, for example to add a new source or change and old source's composition, timing, or owner ship.

Batch Manager:

The Heart of the batch management tier, this component will read the batch process metadata to find out what input it is expecting and from where, and whom should it contact if it doesn't get the expected data. It will gather the data about each instance of

execution from the transformation manager and the business rules engine layers, save it in the extract repository, and expose it through the user feedback engine.

Transformation management:

This tier does all the conversions and calculations necessary to build a fact table record from the extracts that are drawn from the ODS. Normally this is done in three stages, each of which is saved in a stage database.

Source absorption:

The first stage of transformation consists of changing the data type which is mostly string format (since the extracts come in flat text files, excel workbooks, or web services) to the desired date, number, string and Boolean format that is used in database, checking against expected patterns (X9X9X9 for postal code in Canada), recoding (putting "Female" for "F" in gender column), and cleansing (changing a "X9X 9X9" postal code to X9X9X9 and any other to "Invalid". The results, saved in stage one tables, will follow the layout of extracts that were drawn from the providers.

Data Preparation:

The second stage is to do intra-table derivation of new columns (interval between arrival and visit by a doctor) or segmentation of column data (24 time groups based on the arrival hour). The results, which are saved in stage two tables, will follow the layout of the original extracts that now have extra columns.

Data Propagation

The third stage will converge the different extracts together to build up records for different fact tables. This is where inter-table derived columns are calculated and added to the data, and different tables are combined with each other to produce new fact records. The result tables differ from the layout of the original extracts and conform to the structure of the fact tables in the data warehouse.
Having this layer in the architecture is a good measure of future extensibility and maintainability of the system. If new fact tables are needed in the future of the structure of current fact table changes, only the converging sub component of the data propagation component needs to change. Everything else will remain as it was.

Audit Management

The main goal for the audit management tier is to add a data quality dimension to every record that is added to the fact table, so that the user can decide how reliable the data is. In order to do that, the audit manager will use the components in this layer to calculate an audit overall quality, completeness, validation, out-of-bounds, extract time stamp, clean time stamp, conform time stamp, ETL master version, and other indicators for the row of

the data that is about to be added to the data warehouse. The main advantage of this architecture is that the audit manager can start with a small number of such indicators and add them as the need for them arises.

Enterprise integrator

Kimball suggests that the data quality engine should be implemented for each ETL separately, because the intensive workload that it puts on the system might slow down the warehouse operation if it is central. If this view is shared by the directors of VCH, the Enterprise integrator is developed so that an enterprise view of data quality can be provided to the authorized users. This way, immediate collection and management of errors at a local level would remain simple; and a real-time view for the whole enterprise can be provided to the high management at an acceptable increase in complexity. The issue, however, remains open for discussion at the time of this report.

Backtrack Manager

Once quality factors for each row of the fact tables are calculated, it is expected that the user would need to backtrack and see how the row in question was formed and what the original data was. This requires a special mechanism that would account for the convergence and inter-table derivations, since those operations could potentially mask the origin of data.
To implement this functionality, original natural keys are maintained in the fact table record whenever convergence occurs. The backtrack manager would use this information to navigate the user from the fact data upstream to the extract data, so that the origins of the problems are revealed.

ETL Monitor

The overall ETL process needs a service to provide information on its usage and operation. The ETL monitor component is responsible for that service. The usage, workload, and throughput of different components of the system are evaluated and reported through this component, and it has a configuration manager that could tune the priority and accessibility for each part of the system, by which the granularity of audit details can be adjusted.
According to CIHI criterion 46, Usability of the data requires a subset of the microdata to be frozen per release. This is the job of the ETL Monitor.

Predictive Data Manager

In depth analysis and discovery test are performed using this tool, PDM, which permits data mining operations on the audit data. The data mining operations are conducted by seasoned subject matter experts and employ the help of statistical models. The results of these discovery tests, for example “the golden records” would later be fed to the business rules engine to add more screening test and to elevate the level of data quality assurance. The PDM tool is also the main service for visual inspection tests.

Business Rules Engine

The business rules engine is the foundation of the data quality framework. It performs different screening tests on the data as in passes through the ETL pipe, records and tags the unusual values, and selects the best action when an error occurs. The business rules engine is the main service for metadata verification tests and assertion tests.

Error Event Metadata

This is a repository of metadata about the various errors that can be detected using the screening tests. For each screening test, there are baseline parameters that are kept here and changed periodically as discovery tests reveal more realistic baselines. It also shows the data that should be logged when a specific error occurs.

Column Screening metadata

The column screening metadata holds the rules that the specific columns in extracts need to comply to, and specify what should happen if they don't. for example, it might be said that the contents of a column must be a member of a finite list of predefined strings, and if a string outside the list is found, it should be replaced by "Invalid" string. Another example might be that the column contents should not fail the spell checker test and if they did, they should be replaced by the first suggestion of the spell checker program.

Structural Screening Metadata

The tests here involve columns from different tables that, for example, enforce referential integrity on the incoming data. An example would be that for every patient present in the visit extract, there must be a row in the demographics table that would supply the demographic info about that patient. Again the selected action that needs to be performed in case of this test failing is also specified in the metadata of the screening tests.

Business Rules Screening Metadata

These rules fall into three different categories: **Simple data rules**, such as stating that an admission date should always be sooner than operation date for the same patient, **Complex Data rules**, such as stating that there can be no visit records for a doctor after that doctor is pronounced to have left the hospital, and **Value rules**, such as the number in a column should not be more than three standard deviations away from the baseline mean of that column, or that the total number of records in an extract cannot vary more than 10% from one month to the next.

Screening Manager:

The Screening manager is the heart of the business rules engine. It is responsible for combining the metadata from screening metadata repositories with the baseline data from

the error event metadata and formulating the screening test on a specific value. After performing the screen, the screening manager will use the screening metadata to figure out what it should do and to log the event based on the information obtained from the error event metadata.

Error Event Manager

The error event manager logs the error thrown by the screening manager, and exposes them for the audit and batch managers to include in the calculations for their indicators. It also exposes the base data for the DQ Services layer, that act as the interface between the system and the different kinds of users it has. The error event manager is actually an event service because other components need to register with it to get notifications when an error they are interested in is caught.

DQ Services

The Data quality system is exposed to the users and other actors as a collection of services. The main services that this layer provides are Monitoring, Profiling, and Reporting.

DQ Monitoring

The Monitoring services provide the user with the resource usage, workload and through put of the components of the system that the user has authorization to view. This service uses the back office of the User feed back engine and the ETL Monitor and integrates them with engines of its own to provide a unique monitoring interface for the whole system that can e exposed through the organization portal. Here, criteria 30 (monitoring database for efficiency) and 57 (monitor database usage) are satisfied.

DQ Profiling

Profiling Services combine the functionality of the PDM tool, the Backtrack Manager, and the Enterprise integrator components with the capability to see and configure the six metadata repositories that are used through out the system and expose them as a single entity to the users. This means that from the users point of view, the metadata is integrated while the current technology might not allow it (see the chapter on metadata in this document). Obviously, the discovery and visual inspection tests are performed through this window, serving the historical comparability characteristics of CIHI framework through criteria 43 (trend analysis), 44 (categorize the problems in comparing data over time), 55 (Change adaptation in database) and 45 (document the change of data over time).

DQ Reporting

The DQ reporting service is the implementation of CIHI quality assessment tools (). It will also serve the Timeliness and Usability dimensions of the CIHI framework by

providing the user data quality reports at the time of release (criterion 31), scheduling reports through batch manager (criterion 32), and notifying key users of changes (Criterion 52).
DQ reporting service is the main tool to implement the CIHI Relevance dimension. It provides users with feedback (Criterion 53), solicits user satisfaction (Criterion 58), and informs them of developments in the field (Criterion 54)

References:

1. Ministry of Labour and Citizen's Services, BC Government Information Resource Management Glossary, http://www.cio.gov.bc.ca/other/daf/IRM_Glossary.asp accessed May 25th, 2008
2. Strong, Diane M., Lee, Yang W., and Wang, Richard Y. Data quality in context *Association for Computing Machinery. Communications of the ACM*; New York; May 1997, 40, 5, 103-110
3. Wang, Richard Y., and Strong, Diane M. Beyond accuracy: What data quality means to data consumers, *Journal of Management Information Systems*; Armonk; Spring 1996, 12, 4, 5.
4. Kimball, Ralph. Mastering Data Extraction, DBMS Online, June 1996. http://www.dbmsmag.com/9606d05.html, accessed May 24th, 2008
5. Olson, Jack E. Data Quality, 2003. Elsevier
6. Wikipedia. Data Quality, http://en.wikipedia.org/wiki/Data_quality accessed May 27th, 2008
7. Wikipedia. Data Profiling, http://en.wikipedia.org/wiki/Data_profiling accessed May 27th, 2008
8. Mundy, Joy; Kimball, Ralph. The Microsoft Data Warehouse Toolkit. 2006, Wiley.
9. Kimball, Ralph. Meta Meta Data Data. DBMS, March 1998. http://www.dbmsmag.com/9803d05.html accessed May 28th, 2008
10. CIHI, The CIHI Data Quality Framework, 2005
11. New Zealand Health Care Quality Management Framework, 2006
12. Building a data Quality Management Framework for Ontario, 2006
13. Statistics Canada's Quality Management Framework, 2002
14. Kimball, Ralph. An Architecture for Data Quality, 2007.